Ον/μο: Μηναΐδης Θοδωρής

AM: 1115201900113

Εργασία 3

Πρόβλημα 1 :

1. Έστω ότι το πλέγμα έχει διάσταση n x n.
Μεταβλητές -> Πλήθος : n^n , και θα τις ονομάσουμε A1,A2,…,An (για την πρώτη γραμμή) , B1,B2,…,Bn(για την δεύτερη γραμμή),Γ1,……..
Πεδία: {1,…,ν} οι τιμές που μπορούν να πάρουν τα άδεια κενά
Περιορισμοί: Να μην υπάρχει ο αριθμός που θέλουμε να βάλουμε στο κενό στην ίδια γραμμή ή στην ίδια στήλη,
Ανάλογα με την πράξη κάθε κλίκας:
Αν η πράξη είναι + : Το άθροισμα των πεδίων της κλίκας αθροιστικά να είναι ίσο με τον αριθμό που έχει επάνω η κλίκα,
Κάθε αριθμός είναι μικρότερος από τον αριθμό του αθροίσματος ,
Αν έχουμε m κενά κελιά πρέπει ο αριθμός που θα τοποθετήσουμε να είναι τουλάχιστον μικρότερος από τον αριθμό της κλίκας μείον το άθροισμα από i = 1 μέχρι m-1 του I. (έστω χ το αποτέλεσμα της πράξης)
Αν έχουμε c συμπληρωμένα κελιά στην κλίκα ο αριθμός που θα τοποθετήσουμε θα μικρότερος από χ μείον τον αριθμό που έχει κάθε συμπληρωμένο κελί.

Αν η πράξη είναι - : Θα πρεπει ο αριθμός ενός πεδίου (οποιουδήποτε από τα δυο) πλην τον άλλον να ισούται με τον αριθμό της κλίκας

Ο ένας αριθμός να είναι μεγαλύτερος της κλίκας και ο άλλος μικρότερος

Αν η πράξη είναι x : Όλοι οι αριθμοί μεταξύ τους όταν τους πολλ/σουμε να ισούται με τον αριθμό της κλίκας

Όλοι οι αριθμοί μικρότεροι του αριθμού της κλίκας

Αν υπάρχουν αριθμοί τοποθετημένοι σε κελιά της κλίκας διαιρώ τον αριθμό της κλίκας με τους αριθμούς των κελιών και παίρνω ως

Αποτέλεσμα τον αριθμό x

Αν υπάρχουν c κενές θέσεις στην κλίκα διαιρώ το x με c (y το αποτέλεσμα). Τελικά ο αριθμός που θα συμπληρώσω θα είναι μικρότερος

ή ίσος του y.

Αν η πράξη είναι / : Θα πρεπει ο αριθμός ενός πεδίου (οποιουδήποτε από τα δυο) δια τον άλλον να ισούται με τον αριθμό της κλίκας,

Ο ένας μεγαλύτερος του αριθμού της κλίκας, ο άλλος μικρότερος,

Αν έχει συμπληρωθεί κάποιο πεδίο με μικρότερο αριθμό από της κλίκας, τότε πολ/ζουμε τους δύο αυτούς αριθμούς και συμπληρώνουμε

το κενό, διαφορετικά διαιρούμε τον αριθμό του συμπληρωμένου πεδίου με τον αριθμό της κλίκας και συμπληρώνουμε αντίστοιχα

2. Θα χρησιμοποιήσουμε τους αλγορίθμους MAC και FC. Είναι οι πιο γρήγοροι αλγόριθμοι για το συγκεκριμένο πρόβλημα .
Είναι σίγουρα πιο αποδοτικοί από τον απλό BT αφού η μια προσφέρει τον πρώιμο έλεγχο και η άλλη ελέγχει συνεχώς αν οι μεταβλητές έχουν διαθέσιμες τιμές να πάρουν.

Πρόβλημα 2 :

Ναι , είναι δυνατόν.

Αν το δούμε σαν πρόβλημα ικανοποίησης περιορισμών:

Βρισκόμαστε σε ένα επίπεδο 300x400.

Μεταβλητές: είναι τα έπιπλα που θέλουμε να εντάξουμε στο δωμάτιο με τιμές το πλάτος και το βάθος τους : Κρεβάτι, Γραφείο, Καρέκλα, Καναπές.

Πεδία: Τα σημεία των γωνιών κάθε επίπλου (Πάνω – Αριστερά , Πάνω – Δεξιά, Κάτω – Αριστερά, Κάτω - Δεξιά) , κάθε σημείο έχει συντεταγμένες (x,y) εκ των οποίων

το x ανήκει στο διάστημα [0,300] και το y στο διάστημα [0,400]

Περιορισμοί: Δεν μπορεί κανένα σημείο να βρίσκεται στο κάτω αριστερά τετραγωνικό, αφού εκεί είναι η πόρτα καθώς και κανένα έπιπλο να ακουμπάει με οποιοδήποτε

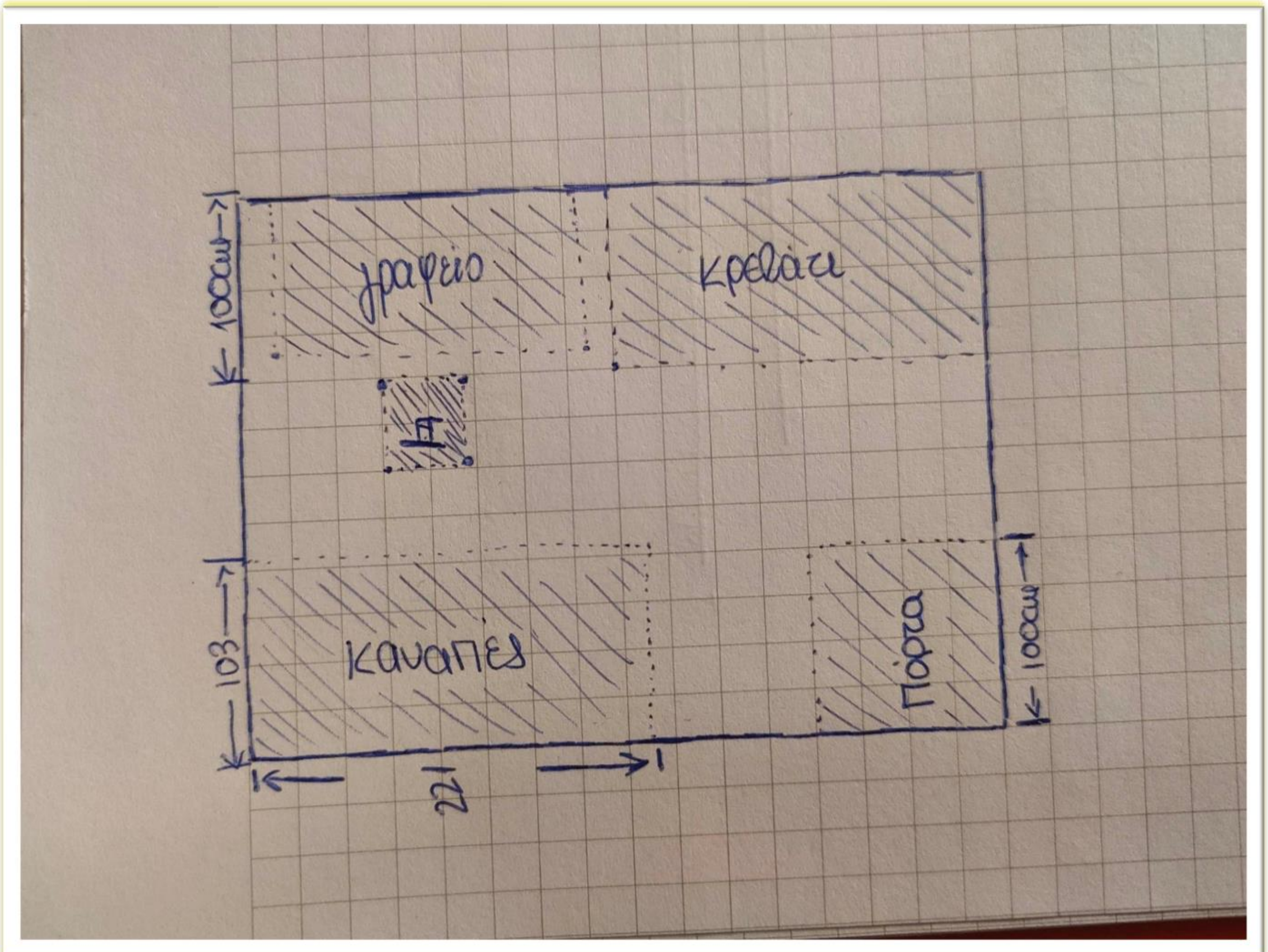
τρόπο πάνω σε αυτό.

Αφού τοποθετηθεί ένα έπιπλο στο χώρο δεν μπορεί κανένα άλλο να βρίσκεται από πάνω του είτε να ακουμπάει σε αυτό (δηλ, δεν γίνεται καμία γωνία να

βρίσκεται μέσα στο ορθογώνιο που δημιουργούν οι γωνίες των επίπλων.

Οι νοητές γραμμές μεταξύ των γωνιών ενός επίπλου δεν επιτρέπεται να πέφτουν μέσα σε άλλα έπιπλα (δηλ, στα ορθογώνια που δημιουργούν οι γωνίες του)

ούτε να ακουμπάνε σε άλλες γραμμές



Διαμόρφωση χώρου με τα έπιπλα :

Πρόβλημα 3 :

Μεταβλητές: Είναι οι διεργασίες Α1,…,Α5

Πεδία: {9:00,10:00,11:00} που μπορεί να πάρει μια μεταβλητή

Περιορισμοί:

- Α1 ξεκινάει μετά την Α3
- Α3 αρχίζει μετά την Α4, πριν από την Α5
- Α2 δεν μπορεί να εκτελείται παράλληλα με Α1 ή Α4
- Α4 δεν μπορεί να πάρει την τιμή 10:00

Γράφος περιορισμών:

```
    A2       ----- A1
  /      \ /
A5 ----- A3 ----- A4
```

AC – 3

Starting algorithm :

#all variables with their possible number for the field

A1 = <9,10,11>

A2 = <9,10,11>

A3 = <9,10,11>

A4 = <9,10,11>

A5 = <9,10,11>

And the queue with the limitations

---------------- |

| A1 > A3 | (1)

| A4 > A3 | (2)

| A5 < A3 | (3)

| A2 != A1 | (4)

| A2 != A4 | (5)

| A4 != 10 | (6)

|-------------- |

And we pop out every element from queue and for every element that its not satisfied , we add it in our queue again

Pop (1) -> A1 = <10,11> -> add (1)

Pop (2) -> A4 = <10,11> -> add (2)

Pop (3) -> A3 = <10,11> -> add (3)

Pop (4) -> all same -> add (4)

Pop (5) -> all same -> add (5)

Pop (6) -> A4 = <11> -> satisfied

Pop (1) -> A1 = <11> , A3 = <10> -> satisfied

Pop (2) -> A4 = <11> -> satisfied

Pop (3) -> A5 = <9> -> satisfied

Pop (4) -> A2 = <9,10> -> satisfied

Pop (5) -> satisfied

Lastly , A1 = 11 , A2 = <9,10> , A3 = 10 , A4 = 11 , A5 = 9

Πρόβλημα 4 :

1. Έγκυρη είναι μόνο η πρώτη πρόταση ,αφού με βάση τον πίνακα 1 είναι η μόνη που για κάθε ανάθεση μεταβλητών ικανοποιείται.

2. Είναι όλες οι προτάσεις ικανοποιήσιμες , αφού με βάση τους πίνακες από κάτω υπάρχουν αναθέσεις μεταβλητών που ικανοποιούν τις προτάσεις.

3. Καμία, αφού είναι όλες ικανοποιήσιμες.

4. Για να υπάρχει τουλ. Ένα μοντέλο που ικανοποιεί μια πρόταση , πρέπει να υπάρχει μια ερμηνεία Ι που να την ικανοποιεί. Από ότι βλέπουμε στους πίνακες

   παρακάτω, για όλες τις προτάσεις υπάρχει ερμηνεία που να τις ικανοποιεί.

5. Η πρώτη πρόταση , αφού για κάθε ανάθεση παρατηρήσαμε ότι η πρόταση ήταν αληθής.

$[(\neg A) \Rightarrow (B \Rightarrow (C \Rightarrow D))]$ x

A	B	C	D	$\neg B$	$(A \wedge (\neg B) \wedge C)$	$x \Rightarrow D$	$\neg[x \Rightarrow D]$	$(\neg A)$	$C \Rightarrow D$	$B \Rightarrow$
t	t	t	t	f	f	t	f	f	t	t
t	t	t	f	f	f	t	f	f	f	t
t	t	f	t	f	f	t	f	f	t	t
t	f	t	t	t	t	t	f	f	t	t
f	t	t	t	f	f	t	f	t	t	t
t	t	f	f	f	f	t	f	f	t	t
t	f	f	t	t	f	t	f	f	t	t
f	f	t	t	t	f	t	f	t	t	t
t	f	f	f	t	f	t	f	f	t	t
f	f	f	t	t	f	t	f	t	t	t
f	f	f	f	t	f	t	f	t	t	t
t	f	t	f	t	t	f	t	f	f	t
f	t	f	t	f	f	t	f	t	t	t
f	t	f	f	f	f	t	f	t	t	t
f	f	t	f	t	f	t	f	t	f	t
f	t	t	f	f	f	t	f	t	f	t f

A	B	$\neg A$	$\neg A \Rightarrow \neg B$	$A \Rightarrow (\neg B)$	$(\neg A) \wedge (\neg A \Rightarrow \neg B) \wedge (A \Rightarrow \sim B))$
t	t	f	t	f	f
t	f	f	t	t	f
f	t	t	t	t	t
f	f	t	f	t	f

Πρόβλημα 5 :

Η παραπάνω πρόταση μπορεί να αναπαρασταθεί με την έκφραση :

Ο λογικός σύνδεσμος είναι =>,

									πρόταση2		πρόταση3	
A	B	C	$\neg B$	$\neg C$	$(\neg A \Rightarrow B)$	$A \vee B$	$\neg A \vee \neg C$	$B \vee \neg C$	$\chi \wedge \psi \wedge z$	$A \vee \neg B$	$A \vee \neg C$	$\chi \wedge \psi \wedge z \wedge \omega$
t	t	f	f	t	t	t	t	t	t	t	t	f
t	f	t	t	f	t	t	f	f	f	t	t	f
f	t	t	f	f	t	t	t	t	t	f	f	f
t	t	t	f	f	t	t	f	t	f	t	t	f
f	f	f	t	t	f	f	t	t	f	t	t	t
f	t	f	f	t	t	t	t	t	t	f	t	f
f	f	t	t	f	f	f	t	f	f	t	f	f
t	f	t	t	t	t	t	t	t	t	t	t	t

Τα σύμβολα σταθερών είναι τα επαγγέλματα , δηλαδή Μάγειρας-Αστροναύτης,

Οι μεταβλητές είναι εγώ και ο Σίφης , άρα έχουμε

Μάγειρας(Σίφης) => Αστροναύτης(εγώ)

Η πρόταση αυτή είναι αληθής όταν:

- Αληθής => Αληθής
- Ψευδής => Ψευδής
- Ψευδής => Αληθής

Εφόσον ξέρουμε ότι το δεύτερο κομμάτι είναι ψευδές , για να ισχύει η πρόταση θα πρέπει να είναι και η πρώτη συνθήκη ψευδής .

Άρα αν «εγώ» δεν έιμαι αστροναύτης , δεν είναι αναγκαστικά και ο Σίφης.

<u>Πρόβλημα 6 :</u>

Σχήμα α (μικρό,μεσαίο,μεγάλο) , (κύβος,δωδεκάεδρο,τετράεδρο)

Σύμβολα σταθερών → Μέγεθος – Σχήμα

Προτάσεις:

Σχήμα(δωδεκάεδρο) ⇔ Μέγεθος(μεσαίο)

Μέγεθος(μεγάλο) ⇔ Σχήμα(τετράεδρο)

Ο σύνδεσμος ⇔ είναι αληθής αν:

Ψευδής ⇔ Ψευδής, και Αληθής ⇔ Αληθής

Για να αποδειχθεί ότι το σύνολο Α έπεται λογικά από την πρόταση Β , πρέπει να δείξουμε ότι για κάθε ερμηνεία Ι => αν Ι(Α) = Αληθής τότε Ι(Β) = Αληθής

Επομένως, αν το σχήμα είναι δωδεκάερδο, είναι μεσαίο μέγεθος. Αν δεν είναι δωδεκάεδρο, δεν είναι μεσαίο.

Και αν το σχήμα είναι τετράερδο, είναι μεγάλο σε μέγεθος. Αν δεν είναι τετράεδρο, δεν είναι μεγάλο.

Άρα , αν το σχήμα είναι κύβος (ούτε τετράεδρο, ούτε δωδεκάεδρο) ,άρα δεν είναι ούτε μεσαίο, ούτε μεγάλο μέγεθος. Το μόνο που απομένει σαν επιλογή είναι το μικρό.

Δηλαδή αν ισχύουν οι προτάσεις Α τότε ισχύει και η Β , δηλαδή $A \vDash \beta$.

Αν το δούμε σαν πρόβλημα ικανοποίησης περιορισμών , ουσιαστικά με έναν αλγόριθμο σαν τον AC3 , από τα domains, θα πηγαίναμε στου κύβου{μικρό, μεσαίο,μεγάλο}

και θα απορρίπταμε τα τελευταία δύο.

Πρόβλημα 7 :

Πρώτα εφαρμόζουμε την μετατροπή σε CNF βηματικά:

1. Απαλοιφή συνεπαγωγών και διπλών συνεπαγωγών
2. Μετακίνηση – προς τα μέσα
3. Επιμεριστική ιδιότητα ν.

1. ¬ (A ⇔ B) ⇔ ¬ ((A ∧ B) V (¬A ∧ ¬B)) ->

   Έστω X = ¬ (A ⇔ B) και Y = ¬ ((A ∧ B) V (¬A ∧ ¬B))

   Άρα X ⇔ Y -> X => Y ∧ Y => X -> (¬X V Y) ∧ (¬Y V X).

   X = ¬(A=>B ∧ B=>A) -> ¬ ((¬A V B) ∧ (¬B V A))

   Τελικά , η αρχική πρόταση γίνεται :

   [((¬A V B) ∧ (¬B V A)) V ((A ∧ B) V(¬A ∧ ¬B))] ∧ [[((A ∧ B) V (¬B ∧ ¬A)) V (¬ (¬A V B) ∧ (A V ¬B))]

2. Μετά τις μετατροπές:
   [((¬A V B) ∧ (¬B V A)) V ((¬A V ¬B) ∧ (A V B))] ∧ [[((A ∧ B) V (¬B ∧ ¬A)) V ((A ∧ ¬B) V (¬A ∧ B))]

3. Δεν χρειάζεται να κάνουμε τίποτα.
   [((¬A V B) ∧ (¬B V A)) V ((¬A V ¬B) ∧ (A V B))]

   [[((A ∧ B) V (¬B ∧ ¬A)) V ((A ∧ ¬B) V (¬A ∧ B))]

Αν B/True τότε:

(Αληθής ∧ A)

Αληθής

Αν A/True :

Αληθής

Αληθής

Αν A/False

Ψευδής

Αληθής

Αν B/False:

Αληθής

Αληθής